IP•Link 1220

Embedded Wireless Module

User Manual

Version 1.0.0

Helicomm

WIRELESS SOLUTIONS•SIMPLIFIED™

Helicomm Inc.

1947 Camino Vida Roble, Suite 109

Carlsbad, CA92008

(760)918-0856

www.helicomm.com

© 2005 Helicomm, Inc.

All rights reserved.

No part of this publication may be reproduced, adapted, or translated in any form or by any means without prior written authorization of Helicomm, Inc.

Information published here is current or planned as of the date of publication of this document. Because we are improving and adding features to our products continuously, the information in this publication is subject to change without notice.

Trademarks

Helicomm, IPWINS, IP•Link, WIN-Gate, and IP-Net are trademarks of Helicomm, Inc. Zigbee is a trademark of the ZigBee Alliance. All other product names mentioned in this publication are trademarks of their respective owners.

Revision and Iteration History

Version	Publication Date	Authors	Summary of Changes and Updates
1.0.0	08/31/2005	CCH	Document Creation

FCC Information

Agency Identification Number	**RF2IPLinkP220**
FCC Notice	"This device complies with Part 15 of the FCC Rules. Operation is subject to the following two conditions: (1) this device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operation."
FCC Labeling Requirement Notice	If the FCC ID is not visible when the module is installed inside another device, the outside of the device into which the module is installed must also display a label referring to the enclosed module. This exterior label can use wording such as the following: "**Contains Transmitter Module FCC ID: RF2IPLinkP220**" "**Contains FCC ID: RF2IPLinkP220**."

Table of Contents

1 Overview 1

2 Module Specifications 2

2.1 IPLink 1220-2033 Interface Pin Definitions 3
2.2 IPLink 1220-2133 Interface Pin Definitions 6
2.3 Special Notes on Interface Pins 8
2.4 Firmware Capabilities Specification 9

3 Theory of Networking Operations 10

3.1 Wireless Networking Topologies 10
3.1.1 Connectivity Topology Versus Routing Topology 10
3.1.2 Star Topology 11
3.1.3 Cluster Tree Topology 12
3.1.4 Peer-to-peer (Mesh) Topology 13
3.1.5 Hybrid Topology 14
3.2 Topology Selection 14

4 Quick Steps in Establishing An IP-Link 1220 Network 16

4.1 Configuring the Master Node 16
4.2 Configuring Non-Master IP-Link 1220 Modules 16
4.3 Forming an IP-Link 1220 Network 17
4.4 Verifying an Established Network 17
4.5 Reconfiguring an Established Network 18
4.6 Special Note: Establishing a Full Mesh Network 19

5 IP-Link 1220 Command Set 20

5.1 AT Command Mode 20
5.1.1 AT Register Table 21
5.1.2 AT Command Error Codes 24
5.2 Binary Mode 25
5.2.1 Generic Frame Format 25
5.2.1.1 Control Header Field 26
5.2.1.2 Link Quality Indicator 26
5.2.1.3 Destination Address Field 26
5.2.1.4 Payload Length Field 27
5.2.1.5 Payload Field 27
5.2.1.6 XOR Checksum Field 27
5.2.2 User Command Request Frame 27
5.2.3 IP-Link 1220 Command Request Code Summary 29
5.2.4 Helicomm Command Response Format 30
5.2.5 Helicomm Data Request Frame 31

5.2.6 Helicomm Acknowledgment Frame 32
5.3 Helicomm Command Synopsis..... 33
6 Terminologies and Acronyms..... 54
7 Mechanical Specification..... 55
7.1 IP-Link 1220-2033 Dimensions..... 55
7.2 IPLink 1220-2133 Dimensions..... 56
7.3 IP-Link 1220-2033 PAD..... 57
7.4 IP-Link 1220-2133 PAD..... 58
7.5 Re-flow Temperature Specifications..... 59
7.6 Solder Paste Recommendations..... 59
8 Ordering Information..... 60
9 Index..... 61

1 Overview

IP-Link 1220 is Helicomm's first embeddable, Surface Mount Technology (SMT) IEEE 802.15.4/ZigBee-compliant wireless module. IP-Link 1220 contains a powerful 8-bit 8051 microprocessor and a 2.4GHz IEEE 802.15.4-compliant RF transceiver. IP-Link 1220 (both 2033 and 2133 models) can operate over 16 channels in the unlicensed 2.4GHz frequency band (or ISM, short for *Industrial, Science and Medical*) across the world.

In addition to its IEEE-standard-based RF and PHY/MAC air interfaces, IP-Link 1220's embedded stack support a wide variety of useful networking features. IP-Link 1220's network support is designed to cover a whole range of application needs, ranging from a simple beaconing network to complicated multi-story full ad hoc networks.

Whether your applications need the robustness and simplicity of IEEE 802.15.4 standard or the versatility of ZigBee Compliance Platform, Helicomm's IP-Link 1220 is the vehicle to enable your applications to the power and cost advantages of standard-based short-range wireless networking. IP-Link 1220 is ideal for a wide range of remote monitoring and control applications such as home control, meter reading, industrial automation, building automation, and security monitoring.

This manual contains vital information about Helicomm IP-Link 1220 embedded wireless transceiver modules. It includes information on how the IP-Link 1220 can be easily provisioned, managed, and integrated into your existing products.

Readers of this document should reference the *IP-Link ZigBee Development Kit 1220* (EZ-NET-1220) documentation, a development tool that facilitates rapid wireless system prototyping using the IP-Link 1220. The IP-Link ZigBee DevKit contains a wealth of detailed diagnostic and pre-built configurations ready to use on a desktop or laptop personal computer. Users will find it a useful tool to help get familiar with the details of IP-Link 1220.

Following is the structure of this document.

- Chapter 2 contains information on the IP-Link 1220 interface, performance and electrical specifications.
- Chapter 3 offers a high-level description of the network operations supported by the IP-Link 1220, and how various network topologies can be configured to meet your application requirements.
- Chapter 4 contains step-by-step instructions on setting up an IP-Link 1220 network. This network configuration guide is followed by a detailed description of the Helicomm Command Set.
- Chapter 5 gives readers definitions and invocation mechanisms needed to develop their own host applications based on IP-Link 1220's flexible networking capabilities.
- Chapters 6 through 8 contain acronyms, mechanical dimensions, manufacturing re-flow specification, and part number information.

2 Module Specifications

Micro-controller (MCU)	MCU Clock Rate	24.5MHz
	FLASH ROM	128 KB
	RAM	8 KB
RF	Frequency	2.4 GHz
	Receiver Sensitivity	-94 dBm
	Air Data Rate	250 Kbps
	Transmit Range	1220-2033: ~150 meters (LOS) 1220-2133: ~450 meters (LOS)
	RF Channels	16 (5MHz)
	Transmit Power	1220-2033: -24 to 0 dBm 1220-2133: -14 to 10 dBm
	Data Encryption	32, 64, 128-bit AES
	Antenna	Chip
	Certification	FCC Part 15, CE
Power Consumption	Transmit/Receive	1220-2033: 55 mA 1220-2133: 85 mA
	Sleep	16 uA
Input/Output	Physical Pins	1220-2033: 62 1220-2133: 70
	Serial	One (1) RS-232 @ 38400 8/N/1
	A-to-D	Three (3) 12-bit ADC Two (2) Comparators
	D-to-A	Two (2) 12-bit DAC
	# of Programmable GPIO	1220-2033: 16 1220-2133: 12
Physical	Dimension (in inches)	1220-2033: 1.6 x 0.7 x 0.2 1220-2133: 1.8 x 0.7 x 0.2
	Dimension (in millimeters)	1220-2033: 41 x 19 x 4 1220-2133: 46 x 19 x 4
	Operating Temperature	-20ºC to +70ºC
	Humidity (non-condensing)	10% to 90%

2.1 IPLink 1220-2033 Interface Pin Definitions

Pin No.	Name	Type	Function Description
1 ~ 8	RF GND	Power	RF Ground pins
9	NC	RF	Not Connected (Note: This pin is reserved for a different antenna option on different SKUs.)
10	RF GND	Power	RF Ground Pin
11	P2.7	Digital I/O	Port 2.7 Digital Input/Output (only available on IP-Link 1220-2033)
12	P2.6	Digital I/O	Port 2.6 Digital Input/Output (only available on IP-Link 1220-2033)
13	P2.5	Digital I/O	Port 2.5 Digital Input/Output (only available on IP-Link 1220-2033)
14	P2.4	Digital I/O	Port 2.4 Digital Input/Output
15	P2.3	Digital I/O	Port 2.3 Digital Input/Output
16	P2.2	Digital I/O	Port 2.2 Digital Input/Output
17	P2.1	Digital I/O	Port 2.1 Digital Input/Output
18	P2.0	Memory Bus Digital I/O	Bit 8 of External Memory Bus (multiplexed mode) Bit 0 of External Memory Bus (non-multiplexed mode) Port 2.0 Digital Input/output
19	P3.7	Digital I/O	Port 3.7 Digital Input/Output
20	P3.6	Digital I/O	Port 3.6 Digital Input/Output
21	P3.5	Digital I/O	Port 3.5 Digital Input/Output
22	P3.4	Digital I/O	Port 3.4 Digital Input/Output
23	P3.3	Digital I/O	Port 3.3 Digital Input/Output
24	P3.2	Digital I/O	Port 3.2 Digital Input/Output
25 ~26	GND	Power	Digital Ground
27	P3.1	Digital I/O	Port 3.1 Digital Input/Output

Pin No.	Name	Type	Function Description
28	P3.0	Memory Bus Digital I/O	Bit 0 of External Memory Bus (multiplexed mode) Bit 8 of External Memory Bus (non-multiplexed mode) Port 3.0 Digital Input/output
29	RX1	UART	UART #1 Data In
30	TX1	UART	UART #1 Data Out
31	RX0	UART	UART #0 Data In (used by IP-Link 1220 firmware)
32	TX0	UART	UART #1 Data Out (used by IP-Link 1220 firmware)
33	TMS	JTAG	JTAG Test Mode, internal pull-up
34	TCK	JTAG	JTAG Test Clock, internal pull-up
35	TDI	JTAG	JTAG Test Data Input, internal pull-up
36	TDO	JTAG	JTAG Test Data Output, internal pull-up
37-38	ANA_GND	Power	Analog ground pins
39	/RESET	Control	Device Reset Open-drain output of internal VDD monitor
40	DAC1	DAC	Digital-to-Analog Converter 1 Voltage Output Range: 0 ~ (V_{REF} -1) mV @ 12-bit resolution
41	DAC0	DAC	Digital-to-Analog Converter 0 Voltage Output Range: 0 ~ (V_{REF} -1) mV @ 12-bit resolution
42	CP1-	Comparators	Comparator 1 inverting input
43	CP1+	Comparators	Comparator 1 non-inverting input
44	CP0-	Comparators	Comparator 0 inverting input
45	CP0+	Comparators	Comparator 0 non-inverting input
46	Vav+	Power	2.7 to 3.6VDC supply
47	AREF		Reference voltage output
48	AIN0.0		ADC 0 Input Channel 0
49	AIN0.1		ADC 0 Input Channel 1

Pin No.	Name	Type	Function Description
50	ANA_GND	Power	Analog ground pin
51	VCC		2.7 to 3.6 VDC
52-58	RF_GND		RF ground pins
59	NC		Not connected
60-62	RF_GNC		RF ground pins

2.2 IPLink 1220-2133 Interface Pin Definitions

Pin No.	Name	Type	Function Description
1 ~ 12	RF GND	Power	RF Ground pins
13	NC	RF	Not Connected (Note: This pin is reserved for a different antenna option on different SKUs.)
14	RF GND	Power	RF Ground Pin
15	P2.7	Digital I/O	Port 2.7 Digital Input/Output (only available on IP-Link 1220-2033)
16	P2.6	Digital I/O	Port 2.6 Digital Input/Output (only available on IP-Link 1220-2033)
17	P2.5	Digital I/O	Port 2.5 Digital Input/Output (only available on IP-Link 1220-2033)
18	P2.4	Digital I/O	Port 2.4 Digital Input/Output
19	P2.3	Digital I/O	Port 2.3 Digital Input/Output
20	P2.2	Digital I/O	Port 2.2 Digital Input/Output
21	P2.1	Digital I/O	Port 2.1 Digital Input/Output
22	P2.0	Memory Bus Digital I/O	Bit 8 of External Memory Bus (multiplexed mode) Bit 0 of External Memory Bus (non-multiplexed mode) Port 2.0 Digital Input/output
23	P3.7	Digital I/O	Port 3.7 Digital Input/Output
24	P3.6	Digital I/O	Port 3.6 Digital Input/Output
25	P3.5	Digital I/O	Port 3.5 Digital Input/Output
26	P3.4	Digital I/O	Port 3.4 Digital Input/Output
27	P3.3	Digital I/O	Port 3.3 Digital Input/Output
28	P3.2	Digital I/O	Port 3.2 Digital Input/Output
29 ~30	GND	Power	Digital Ground
31	P3.1	Digital I/O	Port 3.1 Digital Input/Output

Pin No.	Name	Type	Function Description
32	P3.0	Memory Bus Digital I/O	Bit 0 of External Memory Bus (multiplexed mode) Bit 8 of External Memory Bus (non-multiplexed mode) Port 3.0 Digital Input/output
33	RX1	UART	UART #1 Data In
34	TX1	UART	UART #1 Data Out
35	RX0	UART	UART #0 Data In (used by IP-Link 1220 firmware)
36	TX0	UART	UART #1 Data Out (used by IP-Link 1220 firmware)
37	TMS	JTAG	JTAG Test Mode, internal pull-up
38	TCK	JTAG	JTAG Test Clock, internal pull-up
39	TDI	JTAG	JTAG Test Data Input, internal pull-up
40	TDO	JTAG	JTAG Test Data Output, internal pull-up
41-42	ANA_GND	Power	Analog ground pins
43	/RESET	Control	Device Reset Open-drain output of internal VDD monitor
44	DAC1	DAC	Digital-to-Analog Converter 1 Voltage Output Range: 0 ~ (V_{REF} -1) mV @ 12-bit resolution
45	DAC0	DAC	Digital-to-Analog Converter 0 Voltage Output Range: 0 ~ (V_{REF} -1) mV @ 12-bit resolution
46	CP1-	Comparators	Comparator 1 inverting input
47	CP1+	Comparators	Comparator 1 non-inverting input
48	CP0-	Comparators	Comparator 0 inverting input
49	CP0+	Comparators	Comparator 0 non-inverting input
50	Vav+	Power	2.7 to 3.6VDC supply
51	AREF		Reference voltage output
52	AIN0.0		ADC 0 Input Channel 0
53	AIN0.1		ADC 0 Input Channel 1

Pin No.	Name	Type	Function Description
54	ANA_GND	Power	Analog ground pin
55	VCC		2.7 to 3.6 VDC
56-62	RF_GND		RF ground pins
63	NC		Not connected
64-70	RF_GND		RF ground pins

2.3 Special Notes on Interface Pins

RXD — Receiving data pin for Universal Asynchronous Receiver Transmitter (UART1). Its level should be in accordance with the VDD voltage level. Factory default baud rate is 38400. The default configuration is 8-bit data, no parity, and 1 stop bit.

TXD — Transmitting data pin for Universal Asynchronous Receiver Transmitter (UART1). Its level should be in accordance with the VDD voltage level. The default configuration is 8-bit data, no parity, and 1 stop bit.

/RESET — Module reset signal, low active.

VCC — Supply voltage. All Vcc shall be connected to a power supply in the range of 3.3VDC +/- 10% and less than 20 mVp-p ripple voltage. Higher ripple voltage can significant reduce the transceiver's performance and communication range.

TMS — JTAG Test Mode Select with internal pull-up

TCK — JTAG Test Clock with internal pull-up

TDI — JTAG Test Data Input with internal pull-up. TDI is latched on the rising edge of TCK

TDO — JTAG Test Data Output with internal pull-up. Data is shifted out on TDO on the falling edge of TCK. TDO output is a tri-state driver.

2.4 Firmware Capabilities Specification

Serial Port	Baud Rate	38400
	Configuration	8/N/1
	Maximum Payload over Serial Port	97 Bytes
	Header Length	5
	Checksum	1-byte XOR
	Command Modes Supported	AT Mode (off-line provisioning) Binary Command Mode Binary Data Mode Transparent: RS-232/485 emulation
Networking	Maximum of Network Identifiers	255 (0 ~ 254)
	Range of Node Identifiers	0: Reserved for Network Master 254: Reserved for self-loop back 255: Reserved for broadcast
	MAC Layer Blacklist	8 entries
	Neighbor Table	6-way
	Routing Table	4-way
	RREQ Table	4-way
Sleep Mode	External Wakeup	POR (Power On Reset) Comparators

3 Theory of Networking Operations

IP-Link 1220 can be configured in a number of network topologies to meet different application needs. It allows the users to design a network that best matches their installation conditions and applications' needs. To design a network, it is empirical to understand how each individual IP-Link 1220 should be configured, and what each nodes individual capabilities as well as constrains are.

In this Chapter we discuss the theory of networking operation of IP-Link 1220's networking capabilities to lay the groundwork for later chapters. After reading this Chapter, users should have the system knowledge in assessing, configuring, deploying, and finally fine-tuning their IP-Link 1220 networks in real installations.

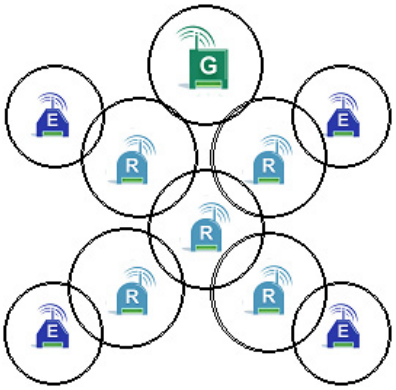
3.1 Wireless Networking Topologies

In this section, we describe the key distinctions between "connectivity" and "routing" topologies to establish the basic framework of wireless network design. We then describe the working details, benefits, and constraints and recommended use case scenarios for the several routing options the IP-Link 1220 supports. This section provides a conceptual platform for readers before they use IP-Link 1220 to build wireless networks.

3.1.1 Connectivity Topology Versus Routing Topology

While the generic phrase *network topologies* suggests wires or cables connecting a host with communicating nodes, wireless communication modules like the IP-Link 1220 use a wireless broadcast medium to communicate. The IP-Link 1220 is a low-power transceiver module optimized for low-cost and low power consumption. So rather than transmitting at high power or having a huge antenna to improve receiver sensitivity, a single IP-Link 1220 transmits at relatively low power (10mW) and utilizes message routing capability to cover a larger area if necessary in some applications. And because of the broadcast nature of wireless transmission, it is important to realize the differences between *connectivity topology* and *messaging topology*.

Connectivity topology refers to the interconnect patterns at the Link level. In a wired network, topology refers to the physical wiring patterns among the nodes. Bus segments or point-to-point Links are some common connectivity topologies seen in Local Area Networks (LAN) or Wide Area Networks (WAN). In contrast, the connectivity pattern of a wireless network is usually visualized as overlapping radio circles or spheres, as illustrated here. The RF sphere implies both range and channelization, which means that nodes with overlapping bubbles are directly connected with one another.



So when considering a connectivity topology, the designer is usually concerned with design parameters such as overall coverage area, nodal density, and the transmission / reception characteristics of the transceiver modules. The characteristics could accidentally change due to varying external conditions and variables such as trucks, walls, trees, and other RF emitters.

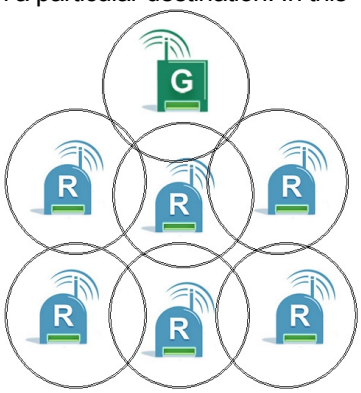
On the other hand, a routing topology is a routing pattern over a multi-hop network. It describes an imaginary wiring diagram, weaving together all network nodes, allowing any arbitrary point to initiate a message (either unicast or multicast) to any fellow node in the network. A routing topology is constrained by the underlying connectivity topology. But for some connectivity topology patterns in which multiple routing options are available (like most wireless networks), selecting the optimal routing topology for your network can be a challenge. Two scenarios are presented here for demonstration.

Scenario 1: Linear Network

Let us examine a linear or "chain fence" scenario, in which any radio can only reach two immediate neighbors in opposite direction. In this extreme case, the choice of routing topology is constrained by the connectivity because there is only one deterministic way of getting a message from point A to point B in the whole network. This topology is common in pipeline monitoring applications and some traffic management and parking meter applications.

Scenario 2: Fully Meshed Network

In this scenario, we increase the size of the RF sphere and make some changes to the relative position. Now one can see that the new connectivity topology offers a wider array of routing options. In this particular diagram, each node will have two or more paths to reach a particular destination. In this case, the routing topology is no longer a simple choice.

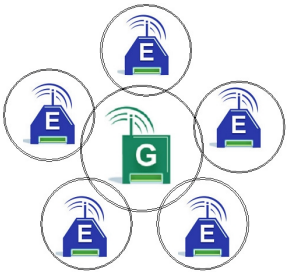


As illustrated in this scenario, routing topology decision for a low-power radio network involves the balance of many design objectives. The wireless network itself is a dynamic system, interacting with its environment incessantly. People movement, intermittent use of electrical appliances, and outside interference sources are all affecting the bubble size. Further complicating the decision process is the design objective to conserve battery consumption for battery-operated devices.

IP-Link 1220's rich wireless routing algorithm is designed to simplify the decision process and expedite the deployment of a reliable, inexpensive wireless infrastructure. Its feature-rich and flexible networking capability aims to provide the network designers with sufficient alternatives and performance margin to easily come to a "just-right" routing topology to adapt to or even overcome the constraints imposed by underlying connectivity topologies.

3.1.2 Star Topology

As its name suggests, a star routing topology is actually a hub-and-spoke system in which data traffic and network commands are routed through a central node, the **Master**. In this routing topology, peripheral nodes require direct radio contact with the **Master**, and interference or the failure of a specific node can render the network less reliable, as each node provides a single point of failure. Especially, the failure of the master node will result in complete system crash. To construct a star network using IP-



Link 1220, only one IP-Link 1220 module needs to be configured as a **Master** node. The remaining IP-Link 1220 modules can be programmed as an **End** node.

The most significant benefit of a star routing topology is its simplicity. The simplicity translates into very low-overhead protocol implementation, much lower overall device cost, very low-overhead routing information, and ease of administration. The central **Master** node can also assume many administrative roles such as certificate authority for authentication, or remote management gateway.

However, the simplicity comes with a price of flexibility. Because of the requirement to put every single end node within the reach of the **Master** node, the overall network coverage is limited. And star topology networks cannot scale up easily to accommodate high-density deployment. The concentrated message routing towards the **Master** node can easily create a hot spot and lead to congestion, packet loss, and performance degradation, depending on the data traffic profile.

The star topology is by far the most common architecture deployed today, and it is well suited for a variety of remote monitoring and control applications that do not need or cannot afford the cost and complexity overhead of a more sophisticated network topology.

3.1.3 Cluster Tree Topology

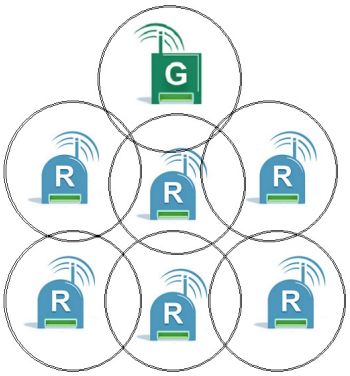
In a nutshell, cluster tree is an aggregation of multiple star topologies. Several stars are chained together to span a much wider area, just like branches and sub-branches in their botanical counterparts. Also known as *spanning tree* topology, cluster tree is a natural networking and routing pattern that is widely in use in today's communication networks.

From a technical perspective, a cluster tree is the simplest topology that can really support network-wide multi-hop messaging services. In support of the networking function, a cluster tree must accomplish several key services at the network layer. First, a cluster tree network must provide a dynamic address allocation scheme so that new-coming nodes can attach themselves to the network and be recognized by other existing nodes. Second, each node in the cluster tree topology must make minimal routing decision, based on the addresses of sender and receiver, in support of message forwarding. Finally, a practical cluster tree network must provide configurable span attributes to specify how much resource a wireless network device can dedicate to support a tree topology. These span attributes include maximum level of depths and maximum number of children allowed. For the embedded wireless networking that the IP-Link 1220 modules are designed for, all these capabilities are provided in its embedded firmware for the users' convenience.

The cluster tree topology most noticeably preserves the simplicity of a star topology: low-overhead routing information, low-to-moderate device memory requirement, and, thus, cost. The new multi-hop capability eliminates the limited ranges by low-power RF transceivers. It allows network architects and application developers to cover a wider premise without necessarily relying on high-power, long-range radios. Finally, cluster tree topology is a highly efficient topology for disseminating network-wide one-to-all or one-to-many messages. Its inherent multicast and broadcast support can provide an elegant solution to applications that require such a group communication mechanism.

Cluster tree, however, does not adapt well to dynamic environment. Its routing mechanism is simplistic and cannot adjust itself easily to very volatile RF environment. Careful readers may find out that, over a cluster tree, one and only one message relaying path exists between any pair of source and destination. Any disruption or failure along this deterministic path means a disconnected network. And because this deterministic path travels through nodes in a pre-programmed fashion, the messaging latency can be high despite the fact two communicating nodes are within range.

Cluster tree topologies are best used for range extension in a stable RF environment. They are also valuable in aggregating a multitude of simple, low-volume data sources (such as sensors.) If your applications require relative range extension with modest networking performance, cluster tree can be a simple topology for your consideration.



3.1.4 Peer-to-peer (Mesh) Topology

Peer-to-peer, also known as mesh networking, is a free-form topology designed to be highly adaptive to the environment. Each node in an IP-Link 1220 mesh network is a little router capable of re-assessing its routing decisions to provide the most robust, reliable network infrastructure possible. After configured as a mesh node (RN+ or Master), each IP-Link 1220 is capable of monitoring surrounding RF conditions, neighboring node activities, and end-to-end packet error rate statistics to adjust its local routing decisions on the fly. Such adaptability is extremely valuable to network designs that are facing uncertain or unpredictable Link conditions.

Mesh topology uses both the RF broadcast nature as well as a set of route inquiry and maintenance commands to dynamically update the distributed routing information across the entire network. The mesh protocol supported by IP-Link 1220 is similar to Ad hoc On-Demand Vector (AODV) routing, in which the node originating a message is responsible for establishing a suitable route by querying its immediate neighbors. The route queries process gradually ripples through the network until the destination confirms connectivity and initiates a reply. Such reply now ripples backwards toward the originator, accumulating vital routing statistics along its way. Finally, the originating node receives the most up-to-date route information and makes a routing decision based on that information. The newly computed routing information will age within a certain window and mandate new route computation after it expires to ensure route decision is based on fresh information.

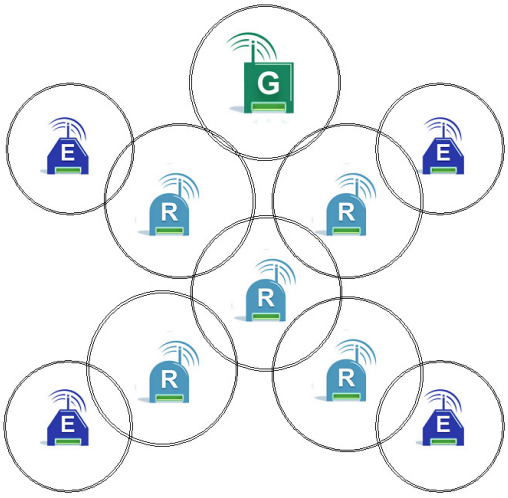
Mesh is ideal for highly unstructured network deployment. When the deployment premise is open and potential interference sources or barriers are anticipated, mesh topology is a reliable way of ensuring wireless connectivity. Especially when deployment density is medium or high, the added redundancy by mesh topologies can add significant design margin and flexibility into the overall networks.

Given its more sophisticated capabilities, however, characterizing and validating a mesh network is more difficult and complicated compared to star or cluster tree networks. Unlike star or cluster tree, a mesh network dynamically adjusts the routing topologies and does not exhibit a fixed, predictable routing pattern. This makes the messaging latency highly dependent on the instantaneous Link quality and difficult to predict. More importantly, a qualitative comparison of mesh algorithms is always a challenging task even for the most savvy network designer.

Network designers usually deploy mesh for applications that require a highly reliable, highly available wireless infrastructure. Mesh networks should also be considered as a means to reduce initial network

setup cost and post-installation maintenance needs by leveraging the self-configuring capabilities embedded inside IP-Link 1220 modules.

3.1.5 Hybrid Topology



In most applications, designers find there are too many design variables to find a dominantly superior topology as we have described. Trade-offs between memory resource requirements, duty cycle, and installation overhead do not make for an easy decision process. In some cases, users will find that a hybrid topology is a compromise that provides balance among the many design constraints.

IP-Link 1220 provides support for hybrid topology by allowing the "grafting" of tree- or star-based peripheral networks onto the robust, highly available mesh backbone network. As we mentioned earlier, a Master or RN+ node is capable of forming a mesh topology that has many desirable features. Nevertheless, cluster tree and star topologies have also good properties, such as deterministic latency, fixed routing pattern, and simple operation. By combining different classes of devices in your network, you will be able to take advantage of the strengths of each topology to best fit your application needs.

The most popular use case of a hybrid topology is to construct a tiered network hierarchy. Similar to its Internet counterpart, a hybrid IP-Link 1220 network usually consists of highly redundant inner rings (the backbone network), consisting of RN+ nodes exclusively. The second-tier network consists of another layer of RN+ to further the mesh backbone, or it is comprised of RN- nodes that will support a simple cluster-tree routing algorithm for the benefits of simpler device construction and lower network maintenance overhead. Finally, the outer-most ring of an IP-Link 1220 network is comprised of the primary End Node that requires no routing intelligence or responsibilities. Such a cascading network hierarchy is extremely pragmatic for most user applications and can be extended easily to meet changing networking requirements.

IP-Link 1220 modules provide users with all of this programmability, allowing them to build, expand, and experiment with all possibilities. In the following chapters, we show the readers how IP-Link 1220 can be configured to achieve these network configurations.

3.2 Topology Selection

IP-Link 1220's rich wireless routing algorithm is designed to simplify the decision process and expedite the deployment of a reliable, inexpensive wireless infrastructure. Its feature-rich and flexible networking capability aims to provide the network designers with sufficient alternatives and performance margin to easily come to a "just-right" routing topology to adapt to or even overcome the constraints imposed by underlying connectivity topologies.

Deciding the routing topology of your applications can be very easy with IP-Link 1220. The decision usually needs answers for the following series of questions:

1. Worst-case and average-case connectivity topologies: What type of installation density do your applications call for (e.g., what is the longest and average distance between your devices), and what is the surrounding environment's conditions in terms of RF interference, building structure and moving objects?

2. Evaluate routing alternatives: select from one of the topologies discussed in this chapter. Based on the information from (1), select a core routing topology that meets your design objectives.

3. Fine-tune routing alternatives by selectively upgrading potential weak spots and balancing against power/resource design constraints.

4 Quick Steps in Establishing An IP-Link 1220 Network

In this chapter we provide a simple guide to forming an IP-Link 1220 network. The generic flow of building an IP-Link 1220 network consists of a series of steps provisioning the Master Node and non-Master nodes and making them recognize one another. The configuration procedure discussed in this chapter is based on those AT Mode or Binary Mode commands detailed in Chapter 5. This chapter also provides tips on verifying the connectivity of a newly formed network and describes procedures users should follow to reconfigure a network.

4.1 Configuring the Master Node

Configuring a Master Node is the first step in building your IP-Link 1220 network. The Master Node is the central coordinator of an IP-Link 1220 network. It is responsible for overall network parameters (e.g., maximum depth and maximum number of children for each node), accepting new IP-Link 1220 modules, and assigning network addresses to enable network-wide routing.

Users should define the following parameters to properly set up the Master Node:

RF Attributes:
- RF Channel Selection (AT Register 114)
- RF Tx Power (AT Register 112)

MAC Layer Attributes
- MAC Layer Network Identifier (AT Register 190)
- MAC Layer Node Identifier (AT Register 191)
- Black List Table (optional; see Command Synopsys on page 33)

Master-only Network Layer Attributes
- Network Layer Network Identifier (AT Register 151)
- Network Layer Node Identifier (AT Register 152)
- Node Type (AT Register 150)
- Maximum number of Children Nodes (AT Register 153)
- Maximum Tree Depth (AT Register 154)
- Routing Algorithm (AT Register 158)

4.2 Configuring Non-Master IP-Link 1220 Modules

After the Master node is configured, setting up the rest of the network is simple. The majority of the configuration for non-Master nodes concerns “matching” or “following” the basic network attributes set by the Master node. After the joining process is complete, the remaining attributes will be negotiated and furnished over the air transparently.

When setting the Node Type attributes, users should be aware that RN+ and RN- can be used as extended attachment points in the network. So if your application requires fairly wide spatial coverage, consider turning a few modules into RN+ or RN- to allow easy cascading and wider RF coverage of your final network.

For a non-Master IP-Link 1220, users should define the following parameters to set up the modules:

RF Attributes:

RF Channel Selection (AT Register 114): Must match Master's RF Channel Selection
RF Tx Power (AT Register 112)

MAC Layer Attributes

MAC Layer Network Identifier (AT Register 190): Must match Master's MAC Layer Network Identifier
MAC Layer Node Identifier (AT Register 191): Must be unique
Black List Table (optional; see Command Synopsys on page 33)

Master-only Network Layer Attributes

Node Type (AT Register 150): can be either an RN+, RN-, or RFD.

4.3 Forming an IP-Link 1220 Network

After all modules are configured, follow the following steps to form an IP-Link 1220 network:

1. Enable the Master: Always turn on the Master first. The recommended usage is to finish provisioning the Master Node and follow with an ATW command to commit the provision data. In this way the Master Module will perform a soft reset automatically and start waiting for the non-Master node to join the network.

2. Add Routing Nodes: For deployment in which you do not care about network topology and formation, you can add any non-Master nodes in an arbitrary fashion. After completing the provisioning process on non-Master nodes, commit the provision data to non-volatile memory via **ATW** command. After the reset, a non-Master node will try to look out for any in-range network attachment points, which is either Master or attached RN+/RN-, and will attach itself to the network on a best-effort basis. In order for a non-Master node to join, the Master node should be programmed to accept join requests momentarily (CCH).

3. If a newly provisioned non-Master node cannot find an attachment point, it will go into dormant mode periodically and re-initiate the joining process.

4. Control the topology and formation: If you wish to maintain a certain topological formation, two approaches are possible. First, you can physically arrange the modules to restrict the attachment points that a new module can find. A more deterministic way to accomplish this is to take advantage of the Black List Table feature in IP-Link 1220. You can provision a specific Black List Table into individual modules so each module will only converse with certain neighbor modules. This places a software filter on a module so it randomly attaches to any module it finds in range.

4.4 Verifying an Established Network

There is more than one way to verify the IP-Link 1220 network forming process. Below we list some of the quickest methods.

1. Examine Network Layer Attributes: When a module successfully makes a "handshake" with the Master Node and joins the network, it receives critical network layer attributes from the Master. You can use both the AT Mode and Binary Mode, as described in Chapter 5, to review these newly acquired attributes. For modules that have successfully completed the necessary joining process, key attributes such as Network Layer Node Identifier should be changed from the factory default unassigned state. For Binary Mode commands, you can take advantage of the loopback address (Node Identifier 254) to issue a binary command to verify critical attributes.[1]

2. "Ping" tests: If all network attributes appears to be valid, refer to Chapter 5 for some end-to-end built-in commands to verify the network connectivity. To be able to issue these command, you must follow the Command Frame specification and have the most addressing information available to accurately pinpoint the destination module. You can issue either single-shot or continuous probes to the destination module, similar to the familiar **ping** command on TCP/IP networks.

We also recommend that you refer to Helicomm's DevKit tool to streamline the verification process.

4.5 Reconfiguring an Established Network

From time to time users may find it necessary to reconfigure their networks. The reconfiguration process may involve moving modules around, changing their node type, and possibly the entire hierarchy. This section describes how to reconfigure your network, including some of the limitations imposed by the current design.

All the RF, MAC Layer, and Network Layer attributes are preserved inside IP-Link 1220's non-volatile memory. Therefore, all modules in an established network can survive power loss or low battery without necessitating the rebuilding of the entire network. However, the non-volatile nature of these critical network configurations prevents the connectivity topology from changing. So the hierarchical relationship – parent/child or attachment point – remains stationary once the network is formed.

For an RFD that lacks the dynamic route discovery capability, the connectivity topology is used as its default routing topology. For instance, if RFD cannot find its parent, it will no longer be able to communicate with the rest of the network. This is known as the *orphaning RFD* problem. [2] So users must pay special attention to rearranging the module's location to prevent from accidentally putting RFDs out of range from its parents.

To totally rebuild a network topology, every single module's network layer attributes must be purged and be reintroduced into the new network using the joining process described above. You can use the "Reset to Factory Default" commands (in either AT or Binary Modes) to reset the module.

[1] In future releases, Helicomm will provide more extensive state-specific commands to simplify this examination and diagnosis step.

[2] There has been a solution proposed for the Orphaning RFD problem in the ZigBee v1.1 Network Layer Specification. Helicomm is in the process of implementing this solution and will make it available in a future firmware upgrade.

4.6 Special Note: Establishing a Full Mesh Network

A full ad hoc mesh network is appealing to many users because of its ease of configuration. In this configuration, all nodes are viewed as equals, and each of them will be a "trustworthy" neighbor to any other nodes within its radio contact. And many users prefer to deploy a full mesh network without going through the sequential process of joining each and every device into the network. Rather than assigning Network Layer address one at a time via Master Node, some users choose to pre-configure address information. Pre-configure address assignment works particularly well for full mesh network, since run-time path is established dynamically rather than relying on static parent-child relationship.

1. It is quite straight-forward to configure your IP-Link 1220 devices into a full-mesh-capable device. You should prepare to setup every node with the following common configurations:
   - An identical RF Channel
   - An identical MAC Layer Network Identifier (from 0 to 255)
   - An identical NWK Layer Network Identifier (you can use the MAC Layer Network Identifier)
2. Now provision a unique MAC Node Identifier into each module. The unique Node Identifier can be selected from the range of 0 to 253. Then apply this MAC Layer Node Identifier to be the NWK Layer Node Identifier, too. Note that Node 0 in a full mesh network does not have any supremacy over other nodes any more. A full mesh network can operate even without Node 0.
3. Turning on devices: For a full mesh network, devices can be turned on at any arbitrary order.
4. Validating connection: It is strongly recommended that you "walk" the entire network from any node that has an external connection that accepts Helicomm's Binary Mode Command Set. For example, you can hook up a Personal Computer to any node and start querying the entire crew in the network. You can run such a "scan" continuously over an extended period to develop some ideas on your deployment environment as well as the network's stability.

5 IP-Link 1220 Command Set

Helicomm IP-Link supports two categories of external command sets. One is the familiar AT command set that is similar to those supported by Hayes-compatible modems. The second category of commands consists of binary instructions that enable a host processor to use IP-Link 1220 as a wireless network interface.

Application developers usually use AT command set to query and set attributes on a standalone module. After the configuration completes, application software can then invoke a binary command set to issue commands and exchange data packets across the wireless network.

Based on these two command set categories, IP-Link1220 supports two modes when it communicates to the outside applications: **AT Mode** and **Binary Mode**. When IP-Link 1220 powers up, it defaults to the binary mode. User issues special escape sequence to switch into AT Mode, and another special AT command to switch back into data mode.

This chapter is organized as follows:

- Section 5.1 presents the AT command set and detailed definitions on IP-Link 1220's S Register definitions.
- Section 5.2.1 introduces the structure of IP-Link 1220's generic frame format and field definitions.
- Sections 5.2.2 through 5.2.6 give detailed descriptions of the four types of command frames supported by IP-Link 1220.
- Section 5.3 provides detailed information on every command request and its corresponding responses.

5.1 AT Command Mode

IP-Link 1220 provides a host of AT commands to allow easy configuration of key attributes of an IP-Link 1220 module. The following texts describe the AT commands, their parameters, and the responses. You can use any terminal emulation utility or UART communication library on a particular host platform to issue these AT commands to IP-Link 1220.

AT String	Purpose	Parameter	Return String
+++	Escape sequence into AT Mode	N/A	Successful: no return value; returns O when a second "+++" is issued Error: Exxx
- - -N-	Escape sequence into transparent Mode	N = 0 ~ 253, 255 in decimal	N/A
===	Switch to Binary Mode	N/A	N/A

AT String	Purpose	Parameter	Return String
AT#n\r	Set MAC Layer Network Identifier	n = 0 ~ 255	Successful: O Error: Exxx
AT@n\r	Set MAC Layer Node Identifier	n = 0 ~ 254	Successful: O Error: Exxx
ATSxxx?\r	Query Register Value	xxx: S register index (in decimal)	Successful: O Error: Exxx
ATSxxx=yyy\r	Set Register Value	xxx: register index (in decimal) yyy: register value (in decimal)	Successful: O Error: Exxx
AT/$\r	Get IEEE MAC Address	N/A	LongMac=0xhhhhhhhhhhhhhhh
AT/B\r	Get module firmware built timestamp	N/A	Month dd yyyy hh:mm:ss
AT/#\r	Get MAC Layer Network Identifier	N/A	MacNetID=n
AT/@\r	Get MAC Layer Node Identifier	N/A	ShortMacAddress=n
AT/S\r	Query All Register Values	N/A	S100=aaa S101=bbb S102=8 … S230=x
AT/V\r	Query Module Firmware Release Number	N/A	a.b.c
ATW\r	Write Back Settings	N/A	Successful: O Error: Exxx
ATR\r	Restore Default Settings	N/A	Successful: O Error: Exxx

5.1.1 AT Register Table

In this section we present a table of IP-Link 1220 S Registers and valid range for each register location. These register entries can be read and set through the commands described in the previous section. The exact Register indexes and acceptable input values are summarized in the table below.

For maintenance reasons, some of these S Registers should not be modified and are only displayed for informational purpose. These entries are labeled as “Reserved” under the field “Access Type.” Readers are strongly advised **NOT** to modify these S Register settings, or Helicomm cannot guarantee the firmware’s performance.

Register Name	S Register Index (decimal)	Access Type	Purpose	Range (decimal)	Manufacturer Default (decimal)
Module UART Baud Rate	101	R	UART Baud Rate	'2': 38400 bps	2
UART Data Bit	102	Reserved	Number of data bits	N/A	8
UART Parity	103	Reserved	Parity bit	N/A	0
UART Timeout	104	Reserved	Timeout value, in milliseconds, for UART	N/A	5
Buffer Size	105	Reserved	UART Buffer size in bytes		143
RF Send Power	112	R/W	RF Send Power	0: 0 dBm 1: -1 dBm 2: -3 dBm 3: -5 dBm 4: -7 dBm 5: -10 dBm 6: -15 dBm 7: -25 dBm [3]	0
RF Accept and Send buffer size	113	Reserved	RF Accept and Send buffer size		116
RF Channel Register	114	R/W	RF Channel Select Register	0 ~ 15 0: 2.405 GHz 1: 2.410 GHz ... 14: 2.475 GHz 15: 2.480 GHz	0
Node Type	150	R/W		0: Master 1: RN+ 2: RN- 3: RFD 255: unassigned	255

[3] For IP-Link 122x-2133 modules (modules with Power Amplifier), the actual transmit power should be increased by another 10 dBm given the same setting.

Register Name	S Register Index (decimal)	Access Type	Purpose	Range (decimal)	Manufacturer Default (decimal)
Network Layer Node ID	152	R/W		0 ~ 255 254 as self-loop back 255 as broadcast address	255
Maximum Number of Children Nodes	153	R/W	For master only		1
Maximum Tree Depth	154	R/W	For master only		1
Current Number of Children Nodes	155	R			0
Current Level in Cluster Tree	156	R			0
Parent Node ID	157	R			0
Routing Algorithm	158	R/W		0: AODV 1: Cluster Tree 2: CT/AODV	2
Table Expiration Value	159	Reserved			15
Transparent Mode Destination	174	R/W		0 ~ 255	255
Network Layer Node ID, High Byte (reserved for future use)	188	R/W		0 ~ 255	255
Network Layer Node ID, Low Byte	189	R/W		0 ~ 255	255
MAC Layer PAN ID, Upper Byte (reserved for future use)	190	R/W		0 ~ 255	255

Register Name	S Register Index (decimal)	Access Type	Purpose	Range (decimal)	Manufacturer Default (decimal)
MAC Layer PAN ID, Lower Byte	191	R/W		0 ~ 255	255
MAC Layer Node ID, Upper Byte (reserved for future use)	192	R/W		0 ~ 255	255
MAC Layer Node ID, Lower Byte	193	R/W		0 ~ 255	255
Sleep Bank (reserved for future use) [4]	200	R/W	Sleep Duration Factor 1	0x00 ~ 0xFF	0
Sleep Time (reserved for future use) [5]	201	R/W	Sleep Duration Factor 2	0x00 ~ 0xFF	0x0A

5.1.2 AT Command Error Codes

When AT commands execute successfully, IP-Link 1220 firmware returns an upper case "**O**" as a success indication. In the case of execution failure, IP-Link 1220 firmware returns one of the following three error codes to indicate the condition.

Error Code	Error Diagnosis
100	Invalid Command
101	Invalid Register
102	Invalid Value

[4] Support will be available in future firmware release.

5.2 Binary Mode

In Binary Mode, host applications use binary-formatted command and responses to command the local modules as well as communicate to remote nodes across the network. This highlights the key utility of Binary Mode operations compared to AT Mode: to communicate and command remote modules over the network formed by multiple IP-Link modules. That said, there are still shortcut commands in Binary Mode to allow users to quickly perform local module access without forcing the application to go through mode switches. In the simplest terms, Binary Mode and AT Mode have overlapping functionalities and are designed to complement each other.

IP-Link 1220 supports four types of frames in its Binary Mode. **Command Request, Command Response, Data Request, and Acknowledgment.**

To use IP-Link 1220's Binary Mode, a Host Application starts with building **Command Request Frames** to query, configure, and command a remote IP-Link 1220 for networking-related functions. The remote IP-Link 1220 module will automatically return a **Command Response Frame** to notify the execution result to the command-issuing module. The sending application then parses the **Command Response Frame** to take further actions. Some configuration records and sensor information natively supported by IP-Link1220 can also be retrieved using **Command Request** and **Command Response**. These commands are built-in to IP-Link 1220, and these **Commands** cannot be extended or modified by the users.

On the other hand, host applications use **Data Request** and **Acknowledgement Frames** to exchange user-specific data. IP-Link 1220's transport the data frames in an end-to-end fashion without interpreting or manipulating the payload in a **Data Request Frame**. The destination IP-Link 1220 will automatically generate an Acknowledgement Frame to report the reception status of the Data Request Frame. After the network topology is established, **Data Request Frame** is the main interface that application developers can use to exchange information among multiple IP-Link 1220 modules. These frames can also be used to carry user-defined network-wide commands, such that IP-Link 1220 can be extended to support any custom commands users desire.

All these frames can be exchanged from one IP-Link 1220 module to a peer module within the same network. The routing of these frames over any given topology is handled by IP-Link1220's embedded firmware transparently.

5.2.1 Generic Frame Format

All four types of frames – Command Request, Command Response, Data Request, and Acknowledgment – use the same generic frame structure: five (5) bytes of packet header descriptor, 0 to 97 bytes of frame payload, and one (1) byte of XOR checksum at the end of packet.

All IP-Link 1220 binary frames follow the following variable-length frame structure:

Control Header (1)	Link Quality Indicator (1)	Destination Address (2)	Payload Length (1)	Payload (0 – 97)	XOR Checksum (1)

Following is the detailed description of the common packet header descriptor.

5.2.1.1 Control Header Field

Length: one byte

Bit Field Definition:

Bit 7,6,5: Binary Frame Type:

100	command request
110	command response
101	data request
111	data acknowledgement

Bit 4: Reserved for future use. Default to 0.

Bit 3,2,1,0: Packet Sequence Number, modulo 16.

NOTE: This sequence number is specifically designed for user applications. IP-Link 1220's firmware maintains separate sequence numbers for data packets. They are transparent to Binary Mode users.

5.2.1.2 Link Quality Indicator

Length: one byte

Bit Field Definition:

Bit 7: Unused.
Bit 6 ~ 0: A 7-bit hex value representing the incoming packet's Link Quality

Description: The Link Quality Indicator (LQI) is an estimate on the packet's signal integrity. Its value ranges from 0 to 127. The higher the value, the better the signal quality. This estimate is derived from IEEE 802.15.4 PHY layer processing performed by any compliant IEEE 802.15.4 transceiver. Users can use this information to assess the MAC-Link quality of a node's surrounding devices. This estimate can be used in conjunction with RSSI.

5.2.1.3 Destination Address Field

Length: two bytes

Bit Field Definition:

Bit 15 ~ 0: Destination Node's Network Address

Description: 0x00, 0xFE, and 0xFF are all reserved address -- 0x00 for Network Master, 0xFE for loopback (to the sender itself), and 0xFF for broadcast.

NOTE: In version 1.0.00, the upper byte of Destination Address Field is ignored due to resource constrains. Therefore a destination address field containing 0xFF01 is equivalent to a destination address field of 0x01. The upper byte is neglected automatically.

5.2.1.4 Payload Length Field

Length: one byte

Bit Field Definition:

Bit 7~0: Represents the payload length (excluding the 5-byte header and 1-byte XOR checksum) in hexadecimal.

Description: Its valid range should be from 0x00 to 0x61 (decimal 97).

5.2.1.5 Payload Field

Length: variable length from 0 to 97 bytes

Bit Definition: User defined.

Description: The magic number 97 is due to the limitation from IEEE 802.15.4 MAC Layer's maximum payload size.

5.2.1.6 XOR Checksum Field

Length: one byte

Bit Definition:

Bit 7~0: XOR Checksum

Description: The XOR checksum is calculated by perform a byte-wide XOR sum on the entire packet header and payload. If an XOR checksum fails, the frame will be discarded automatically.

5.2.2 User Command Request Frame

In Command Request Frame, an additional byte is used to denote a **Command Code** identifier. Helicomm provides a set of built-in command/responses to allow users to manage and retrieval information regarding the networks as well as the sensor information provided by Helicomm's

hardware solution. Each command code identifier will possess its own syntax for both request and response.

Control Header (1)		Link Quality Indicator (1)	Destination Address (2)	Payload Length (1)	Command Code (1)	Parameters (0 – 96)	XOR Checksum (1)
Command Request (4-bit) **b1000**	Sequence Number (4-bit)						

When composing a Command Request Frame, user applications should supply the following information:

- A four-bit, user-defined packet sequence number: this number will be echoed back in receiver's Command Response Frame.
- Destination node's network address: Combined with the Packet Sequence Number, users can use these two numbers to uniquely match an incoming Response to a pending Command.
- The total payload length (up to 0x60)
- The command code: refer to the table in this section.
- The Command parameter: refer to Command Synopsis
- And the XOR checksum on all the bytes preceding the last

When sending a Command Request Frame, user applications should be ready to manage three possibilities:

1. First, the request completes successfully with the expected Response. In this case, the Command Response Frame will be available in the receiving buffer, and host applications can read the serial port input buffer to gather the Response frame.
2. The second condition is that a remote node returns an error indication. In this case, the end-to-end communication is working properly, but the command request is not accepted. Check command syntax and values to correct such problems.
3. The third condition is potentially a communication failure or invalid local command. For communication failure, users may experience continuing checksum error or timeout. In this case, check your communication quality and environment (e.g., moving the destination node closer to the transmitter, or switch to a simpler network topology.) For an invalid local command, verify that you are using the correct network address to address the local module, and the command is formatted correctly.

5.2.3 IP-Link 1220 Command Request Code Summary

Following is a summary of the Command Request set currently supported by IP-Link 1220, firmware release v1.0.0. Please refer to Command Request Frame Synopsis in Section 5.3 (starting on page 33) for complete, individual command's information.[5]

Command Category	Command Name	Command Code (hex)
Sample and ADC	Get IP-Link 1220 ADC Sample	0x81
	Get IP-Link 1220 RSSI Sample	0x82
	Get IP-Link 1220 Temperature	0x83
Module Settings	Get AT Mode S Register Setting	0x86
	Set AT Mode S Register Setting	0x87
Module MAC Settings	Get MAC layer attributes	0x89
	Set MAC layer attributes	0x8A
	Get MAC Address	0x8B
Power Management	Get Firmware Version Number	0x8C
	Soft Reset Module	0x8F
	Reset to Factory Default	0x90
Module Network Settings	Get Routing Table	0x95
	Set Routing Table	0x96
	Get Neighbor Table	0x97
	Set Neighbor Table	0x98
	Get Children Table	0x99
	Set Children Table	0x9A
	Get RREQ Table	0x9B
	Get Black List Table	0x9C
	Set Black List Table	0x9D

[5] The command set can be subject to change without notice. Please refer to Helicomm's website for the latest documentation and firmware release.

5.2.4 Helicomm Command Response Format

Control Header (1)		Link Quality Indicator (1)	Destination Address (2)	Payload Length (1)	Command Code (1)	Response (0 – 96)	XOR Checksum (1)
Command Request (4-bit) **b1100**	Sequence Number (4-bit)						

Command Response Frame is used to indicate back to the originator the execution results of a Command Request Frame.

If the command executes correctly, first the Command Code field in the Response Frame will echo the original command code. Further, a destination node will return any result in the RESPONSE field. If there is no result to return to the sender a value of 0x00 will be placed in the RESPONSE field

If the command execution fails, the destination node will place a 0xFF into the Command Code field. Further the very first byte in Response field will contain an error code for diagnosis purpose. The following table is a summary of possible error codes.

Error Code	**Value (hex)**	**Comments**
ERROR_XOR_ERROR	0x01	Checksum error
ERROR_SEND_FAIL	0x02	Send failure
ERROR_COMMAND	0x03	Invalid command
ERROR_CMD_PARAM	0x06	Invalid Command Parameter
ERROR_DEST_ERROR	0x07	Invalid Destination Address
ERROR_NET_BUSY	0x09	Network Busy

5.2.5 Helicomm Data Request Frame

Control Header (1)		Link Quality Indicator (1)	Destination Address (2)	Payload Length (1)	Data Payload (0 – 97)	XOR Checksum (1)
Command Request (4-bit) **b1010**	Sequence Number (4-bit)					

In this Data Request Frame, applications can deposit the application-specific data (of up to 89 bytes) into the Data Payload and transmit it to the target receiver. The receivers are expected to return an Acknowledgment Frame.

5.2.6 Helicomm Acknowledgment Frame

Control Header (1)		Link Quality Indicator (1)	Destination Address (2)	Payload Length (1)	Error Code (1)	Error Type (1)	XOR Checksum (1)
Command Request (4-bit) **b1110**	Sequence Number (4-bit)						

If a Data Request Frame is received successfully, the receiver will return a Data Acknowledgement Frame, back to the originator, with 0x00 for both Error Code and Error Type fields. For error conditions, Error Code will be set to 0xFF and error type will contain one of the diagnostic error code shown in the table below.

Error Type	Value (hex)	Comments
ERROR_XOR_ERROR	0x01	Checksum Error
ERROR_SEND_FAIL	0x02	Transmission Failed
ERROR_NET_BUSY	0x09	Network Busy

5.3 Helicomm Command Synopsis

The following sections describe in detail the current command set available on IP-Link 1220. Users can refer to this information to build the command library for their particular host application platforms.

Get IP-Link 1220 ADC0 Sample

Read the sample from IP-Link 1220's ADC0

Command Code

0x81

Description

This command is used to retrieve the sample from IP-Link 1220's built-in analog-to-digital converter (ADC0). IP-Link 1220 has a two 12-bit ADCs at 12-bit resolution. ADC#1 and ADC#0 are available on IP-Link 1220's Pin #40 and #41, respectively, to connect to user's analog signal source.

When returned successfully, the first and second byte should be concatenated together to get the 12-bit ADC sample. The 12-bit ADC sample should be reconstructed using the following C pseudo code:

```
ADC_Value = (ADC_High_Byte << 8 ) | (ADC_Low_Byte);
```

Command Parameters

N/A

Response

ADC High Byte	(1 byte)	the most significant 4 bits of the sample (right-aligned)
ADC Low Byte	(1 byte)	the 8 least significant bits of the sample

Get IP-Link 1220 RSSI Reading

Read IP-Link 1220 RSSI reading

Command Code

0x82

Description

This command retrieves the RSSI value, in dBm, from IPLink 1220. The dBm value is without the sign bit. For instance, a reading of “50” (hex) represents an RSSI value of -(0x50) = -80dBm.

Command Parameters

N/A

Response

RSSI	(1 byte)	RSSI value in hexadecimal, no sign bit

Get IP-Link 1220 Temperature

Read the temperature sample from a remote IP-Link 1220

Command Code

0x83

Description

Issue this command to retrieve the ambient temperature sensed by IPLink 1220. To derive at the actual temperature reading, the following conversion should be applied on the 12-bit sample **S**:

Celcius: ((**S** * 2.4 / 4095) – 0.776) / 0.00286

Farenheit: (Celcius * 1.8) + 32

Command Parameters

N/A

Response

Temperature High Byte	(1 byte)	the most significant 4 bits of the sample (right-aligned)
Temperature Low Byte	(1 byte)	the least significant 8 bits of the sample

Get AT Mode S Register Setting

Get a particular S Register's value under AT Mode

Command Code

0x86

Description

This is a shortcut for getting an S Register's value under AT Mode. It is equivalent to issuing ATSxxx? under AT Mode. The difference is that now this capability now can be used across the network.

Command Parameters

S Register Location	1 byte	S Register index in hexadecimal

Response

S Register Value	1 byte	Value in the requested S Register in hexidecimal

Set AT Mode S Register Setting

Set a particular S Register's under AT Mode

Command Code

0x87

Description

This command can be used to set a remote module's S Register. Users are advised to use this command with caution. Improper use of this command can result in modules unable to communicate to the rest of the network.

Command Parameters

S Reigster Location	1 byte	S Register index in hexadecimal
S Register Value	1 byte	Value for the S Register in hexidecimal

Response

Command Confirmation	1 byte	0x00 (constant)

Get MAC Layer Attributes

Get attributes specific to MAC layer processing

Command Code

0x89

Description

This is a macro command retrieving the destination's MAC layer settings.

Command Parameters

N/A

Response

Reserved	1 byte	0x00, in hex
MAC Network ID	1 byte	MAC Layer Network ID, in hex
MAC Node ID	1 byte	MAC Layer Node ID, in hex
Channel ID	1 byte	RF Channel Selector, in hex

Special Note

In future releases, this command may be subject to format exchange to accommodate address extension.

Set MAC Layer Attributes

Set attributes for MAC layer processing

Command Code

0x8A

Description

This is a macro command used to set up the destination's MAC layer settings.

Users are advised to use this command with caution. Improper use of this command can result in modules unable to communicate to the rest of the network.

Valid MAC Layer Network Identifier ranges from 0 through 255, but two special MAC Layer Node Identifiers have been reserved for special use. ID 255 is reserved for broadcast, and ID 254 is reserved for loopback.

Command Parameters

Reserved	1 byte	0x00
MAC Network ID	1 byte	MAC Layer Network ID, in hex. Range: 0 ~ 255.
MAC Node ID	1 byte	MAC Layer Node ID, in hex. Range: 0 ~ 253.

Response

Command Confirmation	1 byte	0x00 (constant)

Special Note

In future releases, this command may be subject to format exchange to accommodate address extension.

Get MAC Address

Get MAC layer hardware address

Command Code

0x8B

Description

This command retrieves an IP-Link 1220 module's IEEE 64-bit MAC hardware address. For IP-Link 1220, this attribute is unused.

Command Parameters

N/A

Response

MAC Address	8 bytes	64-bit IEEE MAC address, MSB first

Get Firmware Version Number

Get release number of IP-Link 1220 module firmware

Command Code

0x8C

Description

This command retrieves the firmware release number on the destination IP-Link 1220 module.

Command Parameters

N/A

Response

Major	1 byte	Major release number, in hex
Minor	1 byte	Minor release number, in hex
Revision	1 byte	Revision number, in hex

Soft Reset

Reset an IP-Link 1220 module

Command Code

0x8F

Description

This command triggers a soft reset of the destination IP-Link 1220. The destination module will retain all its network settings and be able to communicate with the rest of the network after this soft reset.

Command Parameters

N/A

Return Values

Command Confirmation	1 byte	0x00 (constant)

Reset to Factory Default

Reset all module attributes to factory default

Command Code

0x90

Description

This command restores the factory default settings on the destination IP-Link 1220 module.

After the reset, the destination IP-Link may need to be re-programmed with key communication attributes before it can connect with existing wireless network.

Input Parameters

N/A

Return Values

Command Confirmation	1 byte	0x00 (constant)

Get Routing Table

Retrieve Routing Table entries

Command Code

0x95

Description

This command retrieves the entire routing table entries from the destination IP-Link 1220 module.

Currently the routing table supports up to 4 entries. Each entry consists of five bytes with the following information:

Each Routing Table entry consists of the following information:

Field	Length	Description
Destination	1 Byte	Network Layer Node ID
Status	1 Byte	0=Active 1=Discovery underway 2=Route failed 3=Route expired
Cost	1 Byte	Routing cost; max = 16
Next Hop	1 Byte	Next Hop's Node ID
Time To Live	1 Byte	Time until expiration, in seconds.

Command Parameters

N/A

Response

Routing entry 0	5 Byte	See above for field definiton
…	…	…
Routing entry 3	5 Byte	See above for field definiton

Special Note

In future releases, this command may be subject to format exchange to accommodate address extension.

In future releases, the size of this table may be subject to adjustment.

Set Routing Table

Program Routing Table entries

Command Code

0x96

Description

This command sets the entire routing table entries for the destination IP-Link 1220 module.

Currently the routing table supports up to 4 entries. Each entry consists of five bytes with the following information:

Refer to **Get Routing Table** command in the previosu page for Routing Table entry definition.

This command is a variable-length command. That is, it can accept a partial routing table. All unspecified entries on the destination module will be default to 0xff.

This command can only affect the routing behaviors of RN+ nodes in the network. And IP-Link 1220 firmware control routing table entries expiration, so the entries programmed into a remote module's routing table is only temporary.

Users are advised to use this command with caution. Improper use of this command can result in modules unable to communicate to the rest of the network.

Command Parameters

Routing entry 0	5 Byte	See previous command for field definiton
…	…	…
Routing entry K, K< 3	5 Byte	See previous command for field definiton

Response

Command Confirmation 1 byte 0x00 (constant)

Special Note

In future releases, this command may be subject to format exchange to accommodate address extension.

In future releases, the capacity of this table may be subject to adjustment.

Get Neighbor Table

Retrieve Neighbor Table entries

Command Code

0x97

Description

This command is used to retrieve the Neighbor Table of an Zigbee network node. The Neighbor Table captures all reachable IP-Link 1220 modules surrounding the destination modules.

Currently the routing table supports up to 6 entries. Each entry consists of five bytes with the following information:

Field	Length	Description
Destination	1 Byte	Destination's Network Layer Node ID
Destination MAC	1 Byte	Destination's MAC Layer Node ID
Relationship	1 Byte	0 = Parent 1 = Child 2 = Sibling 3 = Neighbor
Level	1 Byte	Number of levels in a cluster tree
Timestamp	1 Byte	When this entry is established. **Currently reserved.**

Command Parameters

N/A

Response

Neighbor Table entry 0	5 Bytes	See above for field definiton
…	…	…
Neighbor Table entry 5	5 Bytes	See above for field definiton

Special Note

In future releases, this command may be subject to format exchange to accommodate address extension.

In future releases, the capacity of this table may be subject to adjustment.

Set Neighbor Table

Program Neighbor Table entries

Command Code

0x98

Description

This command sets the neighbor table entries for the destination IP-Link 1220 module.

Currently the neighbor table supports up to 6 entries. Each entry consists of eight bytes with the following information:

Refer to **Get Neighbor Table** command in the previosu page for Neighbor Table entry definition.

This command is a variable-length command. That is, it can accept a partial routing table. All unspecified entries on the destination module will be default to 0xff.

This command can set up the neighbor table but does not actually reflect the neighboring RF condition of the subject module. IP-Link 1220 firmware will constantly maintain its neighbor table based on inter-module traffics.

Users are advised to use this command with caution. Improper use of this command can result in modules unable to communicate to the rest of the network.

Command Parameters

Neighbor Table entry 0	5 Bytes	See previous page
…	…	…
Neighbor Table entry K, K< 6	5 Bytes	See previous page

Response

Command Confirmation	1 Byte	0x00 (constant)

Special Note

In future releases, this command may be subject to format exchange to accommodate address extension.

In future releases, the capacity of this table may be subject to adjustment.

Get Children Table

Retrieve Children Table entries

Command Code

0x99

Description

This command retrieves the entire children table entries from the destination IP-Link 1220 module.

Currently the children table supports up to 6 entries. Each entry consists of two bytes with the following information:

Field	Length	Description
Destination	1 Byte	Destination's Network Layer Node ID
Node Type	1 Byte	0=Master 1=RN+ 2=RN- 3=RFD 255=unassigned

Command Parameters

N/A

Response

Children Table entry 0	2 Byte	See above for field definiton
…	…	…
Children Table entry K, K< 6	2 Byte	See above for field definiton

Special Note

In future releases, this command may be subject to format exchange to accommodate address extension.

In future releases, the capacity of this table may be subject to adjustment.

Set Children Table

Program Children Table entries

Command Code

0x9A

Description

This command sets the children table entries for the destination IP-Link 1220 module.

Currently the children table supports up to 6 entries. Each entry consists of two bytes with the following information:

Refer to **Get Children Table** command in the previosu page for Children Table entry definition.

This command is a variable-length command. That is, it can accept a partial children table. All unspecified entries on the destination module will be default to 0xff.

Children Tables are usually initialized when a master/RN+/RN- module accepts the join request from a new IP-Link 1220 module.

This command can set up the children table and its setting is permanent until changed.

Users are advised to use this command with caution. Improper use of this command can result in modules unable to communicate to the rest of the network.

Command Parameters

Children Table entry 0	2 Byte	See above for field definiton
…	…	…
Children Table entry K, K< 5	2 Byte	See above for field definiton

Response

Command Confirmation	1 Byte	0x00 (constant)

Special Note

In future releases, this command may be subject to format exchange to accommodate address extension.

In future releases, the capacity of this table may be subject to adjustment.

Get RREQ Table

Retrieve RREQ Table Entries

Command Code

0x9B

Description

This command retrieves the entire RREQ (or Route Request) table entries from the destination IP-Link 1220 module. This command should be only issued to RN+ nodes in a network to observe their Route Request table dynamics

Currently the RREQ table supports up to 4 entries. Each entry consists of three bytes with the following information:

Field	Length	Description
RREQ ID	1 Byte	Identifier for pending Route Request
Source Address	1 Byte	The originator's Network Layer Node ID
Expiration	1 Byte	Time until expiration

Command Parameters

N/A

Response

RREQ entry 0	3 Byte	See above for field definiton
… …	…	
RREQ entry 3	3 Byte	See above for field definiton

Special Note

In future releases, this command may be subject to format exchange to accommodate address extension.

In future releases, the capacity of this table may be subject to adjustment.

Get Black List Table

Retrieve MAC layer Black List Table entries

Command Code

0x9C

Description

This command retrieves the Black List Table on the destination IP-Link 1220 module. Black List Table is a MAC Layer filtering mechansim that forces a module to ignore messages from those nodes listed on the Black List Table.

Currently the Black List Table supports up to 8 entries. Each entry consists of two bytes with the following information:

Field	Length	Description
Start	1 Byte	Starting MAC Layer Node ID, inclusive
End	1 Byte	Ending MAC Layer Node ID, inclusive

Command Parameters

N/A

Response

Routing entry 0	2 Byte	See above for field definiton
…	…	…
Routing entry 7	2 Byte	See above for field definiton

Special Note

In future releases, this command may be subject to format exchange to accommodate address extension.

In future releases, the capacity of this table may be subject to adjustment.

Set Black List Table

Program MAC layer Black List Table entries

Command Code

0x9D

Description

This command sets the Black List Table entries for the destination IP-Link 1220 module. Black List Table is a MAC Layer filtering mechansim to force a module to ignore messages from those nodes listed on the Blist List Table.

Refer to **Get Black List Table** command in the previosu page for Black List Table entry definition.

This command is a variable-length command. That is, it can accept a partial Black List Table. All unspecified entries on the destination module will be default to 0xff.

Black List Table can be provisioned on any type of nodes. Once set, its effect is permanent until changed.

Users are advised to use this command with caution. Improper use of this command can result in modules unable to communicate to the rest of the network.

Command Parameters

Black List entry 0	2 Byte	See the previous page for field definiton
…	…	…
Black List entry K, K<8	2 Byte	See the previous page for field definiton

Response

Command Confirmation	1 Byte	0x00 (constant)

Special Note

In future releases, this command may be subject to format exchange to accommodate address extension.

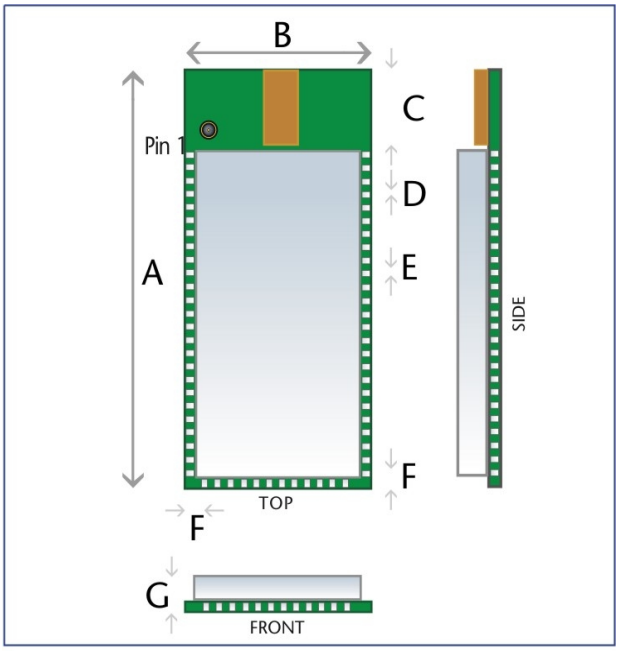
In future releases, the capacity of this table may be subject to adjustment.

6 Terminologies and Acronyms

ADC	**Analog to Digital Converter**
AMR	**Automatic Meter Reading**
CFB	**Cipher Feedback Mode**
CMOS	**Complementary Metal Oxide Semiconductor**
CPU	**Central Processor Unit**
DES	**Data Encryption Standard**
FCC	**Federal Communication Committee**
FSK	**Frequency Shift Keying**
IDE	**Integrated Development Environment**
IF	**Intermediate Frequency**
ISM	**Industrial Scientific Medical**
ISR	**Interrupt Service Routine**
LOS	**Line of Sight**
LPF	**Loop Filter**
LQI	**Link Quality Indicator**
LSB	**Least Significant Bit (or Byte)**
MAC	**Medium ACcess Layer**
MSB	**Most Significant Bit (or Byte)**
PCB	**Printed Circuit Board**
PHY	**PHYsical Layer**
POR	**Power On Reset**
RAM	**Random Access Memory**
RF	**Radio Frequency**
RSSI	**Received Signal Strength Indicator**
RTC	**Real-Time Clock**
RX	**Receive**
SFR	**Special Function Register**
SPI	**Serial Peripheral Interface**
SRAM	**Static Random Access Memory**
SRD	**Short Range Device**
TQFP	**Thin Quad Flat Pack**
TX	**Transmit**
UART	**Universal Asynchronous Receiver/Transmitter**

7 Mechanical Specification

7.1 IP-Link 1220-2033 Dimensions

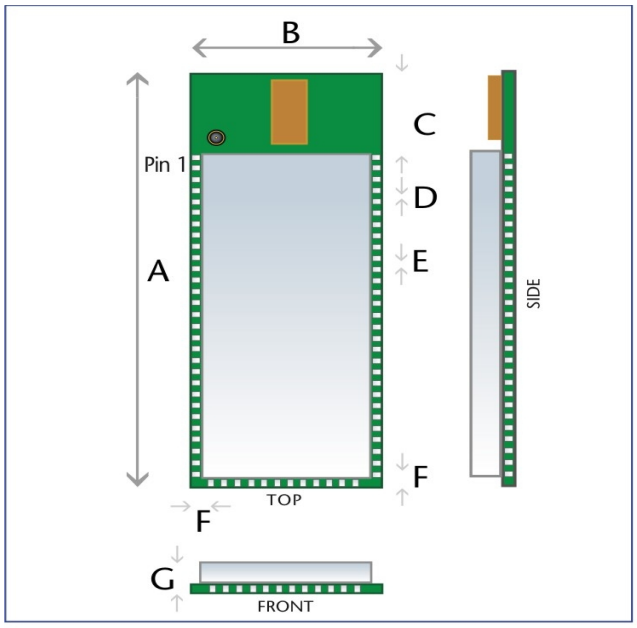


2.4GHz Module Dimensions See Table 2

Table 2: *2.4GHz Module Dimensions*

Module	Ref	Value (mm)	Description
2.4GHz	A	41	Length for Modules with NO PA
2.4GHz	B	19	Width for Module
2.4GHz	C	8.5	Antenna Area
2.4GHz	D	1.27	Distance Between Pin's
2.4GHz	E	0.812	Width of Pins
2.4GHz	F	2.1	Pin Distance from Edge
2.4GHz	G	4	Heigth for Module

7.2 IPLink 1220-2133 Dimensions

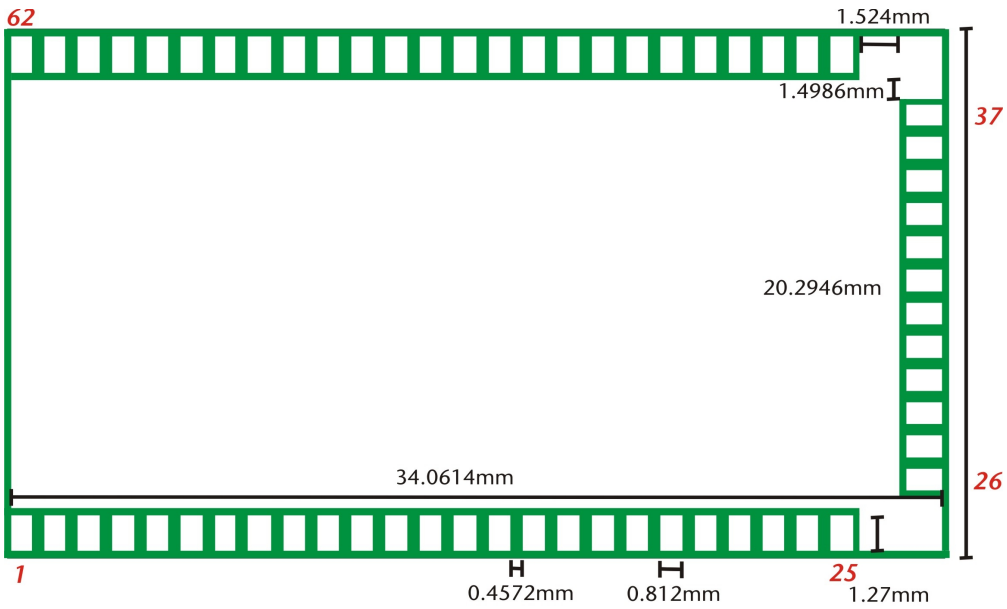


2.4GHz PA Module Dimensions See Table 3

Table 3: *2.4GHz PA Module Dimensions*

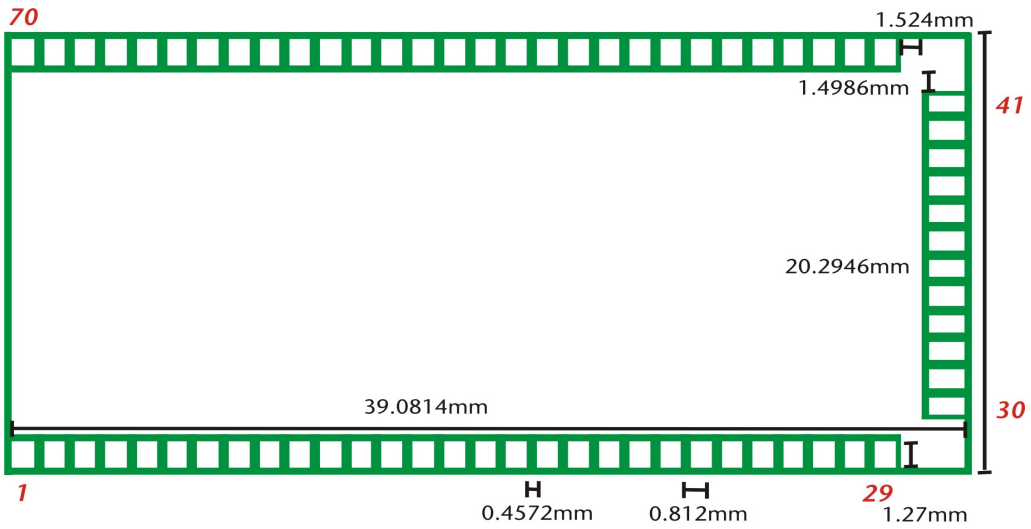
Module	Ref	Value (mm)	Description
2.4GHz	A	46	Length for Modules with PA
2.4GHz	B	19	Width for Module
2.4GHz	C	8.5	Antenna Area
2.4GHz	D	1.27	Distance Between Pin's
2.4GHz	E	0.812	Width of Pins
2.4GHz	F	2.1	Pin Distance from Edge
2.4GHz	G	4	Heigth for Module

7.3 IP-Link 1220-2033 PAD



F1220 Foot Print

7.4 IP-Link 1220-2133 PAD



F1220 PA Foot Print

7.5 Re-flow Temperature Specifications

We recommend low temperature lead-free solder paste rated at 118ºC.

	Ideal (ºC)	Maximum (ºC)
Maximum Re-flow Temperature	118	180

7.6 Solder Paste Recommendations

We recommend low temperature lead-free solder paste rated at 118ºC.

	Alloy Composition	*Solidus (ºC)*	*Liquidus (ºC)*	*Shear MPa*
Johnson Alloy #806	In/48Sn (e)	118	118	

8 Ordering Information

You can contact Helicomm and our resellers for additional modules or develop kit to grow your network. Please specify Product Part Number: **IP-Link 1220-2033 or IP-Link 1220-2133.**

A six-node IP-Link 1220-2133 Development Kit with USB connector, demo sensors, and network management tool can be purchased to jump-start your first experiences with Helicomm's networking technologies. To order the Development Kit, Please specify Product Part Number: **EZDK 1220PA.**

9 Index

A

AT Mode....37, 38

D

Development Kit....
  Part Number....60

I

IP-Link 1000....
  AT Command Mode....20
  AT error code....24
  AT Registers....21
  Binary Mode....25
  Command Set....20
IP-Link 1000 Frame....
  Acknowledgement....32
  Command Request....27
  Command Response....30
  Data Request....31
  Generic....25

M

MAC Layer....40
Master Node....
  configuring....16
  configuring non-master....16

N

Network....
  Forming....17
  Reconfiguring....18
  Verifying....17

T

Table....
  Black List....52, 53
  Children....49, 50
  Route Request....51
Topology....
  Cluster Tree....12
  Connectivity....10
  Hybrid....14
  Peer-to-peer....13
  Routing Topology....10
  Star....11